


Enhancing Older Adults' Well-Being Through QoS-Aware Edge-Cloud eHealth Applications

Md Mahfuzur Rahman^{1,2} 

¹Department of Information and Computer Science, King Fahd University of Petroleum & Minerals (KFUPM),
Dhahran, 31261, Saudi Arabia

²Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS),
King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, 31261, Saudi Arabia
{mdmahfuzur.rahman}@kfupm.edu.sa

Keywords: Older Adults' Healthcare, Edge-Cloud Computing, Quality of Service.

Abstract: Ensuring the well-being of older adults using eHealth applications has become increasingly important. Continuous health monitoring, emergency response, and various personalized applications can improve the well-being of the older adults through intelligent and adaptive data processing. But, the effectiveness of such systems highly depends on the provisioning of efficient computing resources that meet Quality of Service (QoS) requirements (of the applications), including low latency, faster computation, reliability, and security. Leveraging Edge-Cloud Computing can offer a promising solution but an effective strategy is required to understand the Quality of Service (QoS) requirements of applications while offloading in Edge-Cloud Computing. Previous research lacks in focusing predicted behaviour of application workloads (and related QoS) while scheduled on the Edge-Cloud platform. This research addresses this issue by proposing a scalable model facilitating resource allocation based on the specific future requirements of data processing tasks. In this research, an efficient heuristic algorithm is developed to maximize meeting the QoS constraints. The effectiveness of the proposed approach is evaluated through simulations comparing its performance against existing methods, thereby facilitating improved service delivery and user satisfaction.


1 INTRODUCTION

The rapid growth of the aging population presents significant challenges for healthcare systems worldwide. As older adults face increased risks of chronic diseases, cognitive decline, and mobility limitations, the demand for continuous and efficient healthcare services is rising. Traditional healthcare models, which rely heavily on in-person visits and hospitalization, are often insufficient in addressing the real-time health monitoring and intervention needs of older individuals. In response to these challenges, eHealth applications have emerged as a promising solution to provide remote health monitoring, emergency assistance, and personalized care.

Recent advancements in Edge-Cloud Computing have further improved the capabilities of eHealth systems by enabling real-time data processing, intelligent decision-making, and scalable analytics. Edge computing brings computation closer to the data source, reducing latency and improving responsive-

ness, while cloud computing provides centralized storage, long-term analytics, and machine learning-driven insights. However, ensuring Quality of Service (QoS) in such systems remains a critical challenge, as factors like network congestion, device heterogeneity, data security, and real-time processing requirements can impact the overall performance of health monitoring applications. In this paper, we propose a QoS-aware Edge-Cloud eHealth monitoring framework designed to enhance the well-being of older individuals. The framework integrates adaptive resource provisioning to optimize computational efficiency, network performance, and data security.

The proposed approach invents new and efficient heuristic solutions based on predicted resource allocation schemes. Consider, for example, an older people or patient monitoring app (running on a smart phone) that processes Internet of Things (IoT) data obtained from various IoT sources such as wearable physiological sensors, etc. The workload created with this monitoring app can be handled locally by the smartphone but since the smartphone is a power-limited device,

^a  <https://orcid.org/0000-0002-2871-9119>

the computation can be also located in Edge-Cloud environment depending on the QoS preferences. By analysing the collected IoT data, the app can also detect any abnormal behaviour of the patient and subsequently more IoT devices (e.g. IP camera) can be turned on and the obtained video data are then analyzed to monitor and understand the patient's movement more precisely and effectively. In that case, the computation is preferred to be taken place in Edge-/Cloud environment considering the predicted future scalable workload. Essentially, it needs to be decided that which part of the tasks should be offloaded to Edge/Cloud so that smartphone resources can be still utilized efficiently and also high computation requirement can be satisfied by migrating workload to Edge-/Cloud. This migration to either Edge or Cloud also depends on various QoS requirements (e.g. the minimum tolerable latency, privacy of data, etc.). Therefore, in summary the research has the following key goals:

- Design a dynamic computation offloading framework for QoS-aware resource allocation for eHealth data and services.
- Evaluate the suitability of proposed approach through simulations with a simulation tool.

Section 2 describes the related work and section 3 details the design of the dynamic computation offloading framework for QoS-aware resource allocation for eHealth data and services. Section 4 shows the evaluation results, and finally, Section 5 concludes the paper.

2 RELATED WORK

Edge computing has been widely adopted in healthcare systems to enable real-time health monitoring and reduce latency in critical medical applications. Studies such as (Islam et al., 2024) and (Ahmed et al., 2023) propose edge-based architectures for processing vital signs, motion detection, and chronic disease monitoring in older individuals. The cloud component provides additional storage, long-term analysis, and integration with medical institutions. However, these studies primarily focus on edge-based computation without addressing dynamic resource allocation challenges for QoS enhancement.

Meeting QoS requirements in data processing is crucial. Without fulfilling strict QoS requirements, the results may not be presented to the users in a satisfactory manner. Various researchers (Herrera et al., 2020; Külzer et al., 2021) have addressed QoS issues in data processing scenarios. Hoseinyfarahabady

et al. (Hoseinyfarahabady et al., 2019) discusses a cloud scenario, where the disk I/O bandwidth is identified as the potential bottleneck causing QoS violation. Thus, they proposed an instance placement algorithm with full consideration of disk I/O balancing. The authors in (Hoseiny et al., 2021) formulated task scheduling problem as an NP-hard problem and introduced two task scheduling algorithms to allocate IoT workload on edge-cloud environment. The algorithms minimize the computation cost, communication cost, and delay violation and the performance improvement made by those algorithms were compared with the genetic-based algorithm.

Recent research in (Mukhopadhyay et al., 2024) has explored hybrid Edge-Cloud frameworks for remote health monitoring, where computing tasks are intelligently offloaded to either edge or cloud nodes based on network conditions and resource availability. These works highlight the importance of low-latency decision-making but fail to address the impact of network congestion, device heterogeneity, and real-time QoS constraints, which are critical for older adults' healthcare applications. Works in (Louvros et al., 2023) and (Chi et al., 2020) introduce QoS-aware scheduling algorithms that prioritize emergency cases and optimize resource allocation across distributed healthcare networks. However, they rely on static provisioning techniques that do not dynamically adjust to real-time network conditions, leading to potential delays in older adults' healthcare applications.

Moreover, research in (Peng et al., 2023) proposes Artificial Intelligence (AI)-driven resource scheduling that leverages machine learning to predict network congestion and adjust resource allocation dynamically. While this enhances service reliability, the high computational overhead of AI models may pose scalability issues when deployed on resource-constrained edge devices. Righi et al. (da Rosa Righi et al., 2020) used Autoregressive Integrated Moving Average (ARIMA) and Weighted Moving Average to predict IoT load behavior and to anticipate future scaling in and out. Etemadi et al. (Etemadi et al., 2020) used time series prediction model to predict the IoT workload and Xu et al. (Xu et al., 2020) used ensemble learning algorithm to make accurate predictions on IoT workload. Our proposed framework builds upon workload prediction approaches by introducing adaptive edge-cloud resource provisioning that balances latency, computational efficiency, and network performance for real-time older person's health monitoring.

The study in (Rema and Sikdar, 2021) focuses on addressing overcrowding in hospital emergency departments (ED) by employing quantitative methods

for resource planning and deployment. By analyzing patient flow and forecasting demand, hospital administrators can make informed decisions. The study examines 7748 ED arrivals from a Bengaluru hospital, analyzing patient flow during each working shift. Time series modeling techniques, particularly exponential smoothing proposed by Hyndman, were used to generate short-term forecasts. Model validation and residual analysis were conducted to ensure accuracy. Prediction intervals with a 90% confidence level were obtained on a shift-wise basis, allowing for efficient resource reallocation and demand estimation by hospital management. (OpenAI, 2024).

3 PROPOSED SYSTEM

Edge-Cloud system architecture combines the distributed computing capabilities with the scalability and storage potential of cloud infrastructure, creating a versatile framework for IoT applications. At its core, this architecture leverages edge devices or fog nodes situated closer to the data source for real-time processing and decision-making, reducing latency and bandwidth usage. These edge/fog nodes act as intermediaries between IoT devices and the centralized cloud servers, filtering and analyzing data locally before transmitting relevant information to the cloud for further processing or storage. This hierarchical approach optimizes resource utilization and enhances the responsiveness and efficiency of IoT systems, making them suitable for a wide range of applications, from smart cities to industrial automation. Additionally, the edge-cloud architecture offers flexibility in deployment, enabling seamless integration of new devices and services while ensuring data security and privacy through robust encryption and access control mechanisms. Figure 1 shows an architecture related to Edge-Cloud system architecture considered in this research.

3.1 Remote Healthcare Monitoring Application

An eHealth application represents a digital solution that harnesses technology to deliver healthcare services and information remotely. These applications encompass a wide range of functionalities, including telemedicine consultations, electronic health record management, remote monitoring of vital signs, medication management, and health education resources, etc. A Remote Healthcare Monitoring Application is a digital solution designed to facilitate the remote monitoring of patients' health and medical conditions.

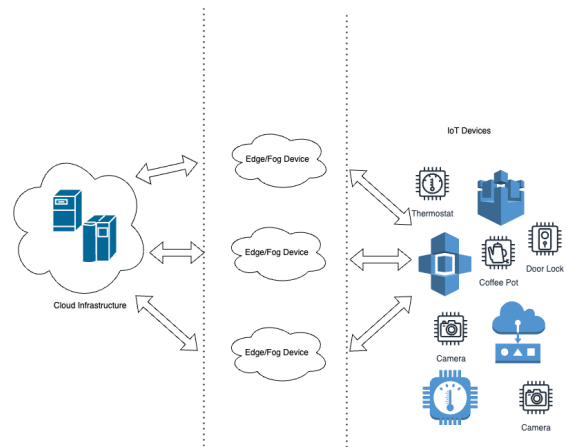


Figure 1: Edge-Cloud Architecture.

This innovative platform leverages technology to enable healthcare providers to monitor patients' vital signs, symptoms, and medication adherence from a distance, thereby enhancing patient care and management. Through the use of various medical devices such as wearable sensors, smart monitors, and mobile applications, patients can conveniently transmit real-time health data to healthcare professionals, allowing for timely interventions and adjustments to treatment plans. Remote Healthcare Monitoring Applications offer numerous benefits, including improved access to care for patients in remote or underserved areas, early detection of health issues, reduced hospital admissions, and enhanced patient engagement and empowerment through active participation in their own healthcare journey. Overall, these applications play a vital role in advancing telemedicine and revolutionizing the delivery of healthcare services by bridging the gap between patients and providers regardless of geographical barriers.

Remote Healthcare Monitoring Applications utilize IoT devices to gather real-time health data from patients remotely. These IoT devices, such as wearable sensors and smart monitors, continuously collect various health metrics like heart rate, blood pressure, and activity levels. This data is then transmitted securely to the monitoring application, where it is analyzed and interpreted. The workload in this context refers to the processing and analysis of the vast amount of health data generated by these IoT devices. In this research, a remote healthcare monitoring application is considered and the related workload is efficiently analyzed to ensure timely monitoring and intervention for patients. Additionally, the historical time-series information of workloads (related to data processing, analysis, and interpretation) are considered to make future workload prediction for providing effective remote healthcare monitoring services.

3.2 Dataset

Workload datasets provide valuable insights into the performance and efficiency of the application, allowing healthcare providers to make informed decisions regarding resource allocation, capacity planning, and optimization strategies. By analyzing workload data, providers can identify patterns, trends, and areas for improvement in the delivery of remote healthcare services. Understanding the workload patterns and demands helps in optimizing the allocation of resources such as computing power, storage, and network bandwidth within the application hosting infrastructure. This ensures that the application can effectively handle fluctuations in workload without compromising performance or patient care.

The dataset, used in this research, was gathered from IoT devices installed in apartments occupied by elderly individuals living alone and was subsequently uploaded to the SSiO platform (Swedish Society for Industrial Organization, 2025). It was collected in compliance with General Data Protection Regulation (GDPR) regulations, ensuring that individual identities cannot be discerned from the data. The data was collected on an apartment-by-apartment basis, with timestamps indicating when IoT events occurred. The dataset spans from 2019 to 2021. This realistic incoming traffic from the SSiO IoT healthcare application system (Swedish Society for Industrial Organization, 2025) is studied, developed and modeled in this research. Figure 2 displays day-long samples for 4 different days extracted from the considered dataset.

3.3 Prediction Model

Preparing a workload prediction model for a healthcare monitoring application involves several key steps to ensure its accuracy and effectiveness. Initially, it requires gathering and preprocessing relevant data sources, including patient demographics, medical history, vital signs, and historical admission records. Next, suitable time-series models are selected based on the nature of the data and the prediction task. These models are trained using historical data to learn patterns and relationships between variables that influence workload fluctuations in healthcare settings. Additionally, the model may incorporate external factors such as seasonal variations, public health trends, and demographic shifts to improve its predictive accuracy. After training, the model is validated using separate datasets to assess its performance and generalization capability. Finally, the SARIMA model is deployed within the healthcare monitoring application, where it continuously analyzes real-time data to

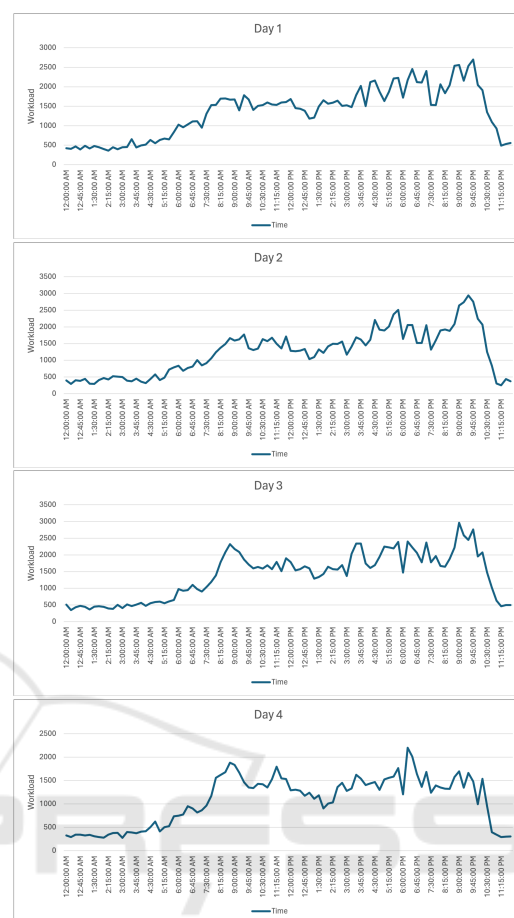


Figure 2: Example Workloads.

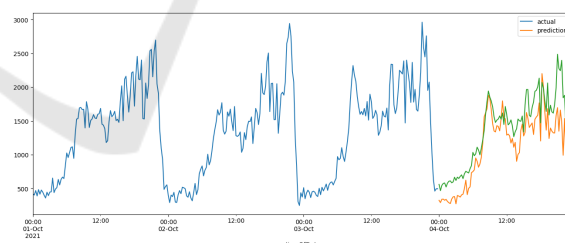


Figure 3: Workload Prediction using SARIMA model.

predict future workload demands (see Figure 3). Regular monitoring and evaluation of the model's performance ensure its reliability and relevance over time, allowing healthcare providers to proactively manage resources and optimize patient care delivery.

3.4 QoS Aware Task Placement (QTP)

The developed task placement algorithm (Algorithm 1) follows a systematic process for task scheduling at a fixed interval μ . Initially, incoming task requests are accumulated into a batch, termed new-

TaskList, and assessed for their Quality of Service (QoS) requirements [Algorithm 1, Line 1-3]. Tasks are categorized based on their QoS sensitivity, including considerations for latency, security, and scalability. If a task's certain QoS requirement crosses a threshold α then it is marked as QoSSensitive to be scheduled properly. If a task is not QoSSensitive then it will be marked as well [Lines 3-9]. These tasks are then merged with existing tasks in taskList for further processing [Line 11]. Workload prediction is conducted for each task until a future time $t+\mu$ (where t is the current time and $t+\mu$ is the next re-evaluation time), distinguishing between Compute-Sensitive and ComputeNonSensitive tasks based on workload thresholds β [Lines 13-16].

Algorithm 1

```

1  Repeatedly scan newTaskList for new task at a
   fixed interval  $\mu$ 
2
3  Characterize each task depending on QoS
   requirements
4
5  Foreach taski in newTaskList:
6      IF taski.QoSRequirements >  $\alpha$ 
7          mark taski as QoSSensitive
8      ELSE mark taski as QoSNonSensitive
9  End For
10
11 taskList  $\leftarrow$  merge(newTaskList, existingTaskList)
12
13 Foreach taskj in taskList:
14     IF predict(taskj.ComputeRequirements,  $\mu$ ) >  $\beta$ 
15         mark taskj as ComputeSensitive
16     ELSE mark taskj as ComputeNonSensitive
17
18 T1  $\leftarrow$  MakeSortedSubList(QoSNonSensitive,
   ComputeSensitive, taskList)
19
20 T2  $\leftarrow$  MakeSortedSubList(QoSSensitive,
   ComputeNonSensitive, taskList)
21
22 T3  $\leftarrow$  MakeSortedSubList(QoSSensitive,
   ComputeSensitive, taskList)
23
24 T4  $\leftarrow$  MakeSortedSubList(QoSNonSensitive,
   ComputeNonSensitive, taskList)
25
26 TN  $\leftarrow$  MergeSubTaskListsSequentially(T2, T3, T4)
27
28 Foreach task tk in T1 :
29     If tk in existingTaskList and tk on Fog
       Platform:
30         offloadingCost(tk, FogToCloud) <  $\gamma$  :
31             success  $\leftarrow$  schedule tk on Cloud
               Platform
32             if success equals false :
33                 add tk at the end of TN
34         ElIf tk in newTaskList:
35             success  $\leftarrow$  schedule tk on Cloud Platform
36             if success equals false :
```

```

37         add tk at the end of TN
38
39 Foreach task tk in TN :
40     success  $\leftarrow$  schedule tk on Fog Platform
41     if success equals false :
42         success  $\leftarrow$  schedule tk on Cloud Platform
43     if success equals false :
44         schedule tk on local machine
45
46 Consider taskList as existingTaskList for next
   interval
```

Tasks are further organized into four sublists (T₁, T₂, T₃, T₄) according to their QoS and Compute sensitivity, with T_N representing a special sublist formed by merging T₂, T₃, and T₄ [Lines 18-26]. Allocation decisions prioritize QoSNonSensitive but Compute-Sensitive tasks for Cloud platform which also considers the cost threshold γ due to the task offloading from Fog to Cloud platforms [Lines 28-33]. Newly arriving tasks are allocated directly to the Cloud platform to address compute-sensitive requirements [Lines 35-37]. Finally, tasks in the T_N special sublist are scheduled sequentially, with preference given to Fog platforms unless QoS constraints necessitate Cloud platform allocation. Tasks unable to be scheduled remotely are assigned to the local machine [Lines 39-44]. At the conclusion of the algorithm, taskList is updated as existingTaskList for scheduling in the subsequent interval [Line 46]. This structured approach ensures efficient task allocation while accommodating diverse QoS and compute requirements within the Edge-Cloud environment.

4 ASSESSMENT

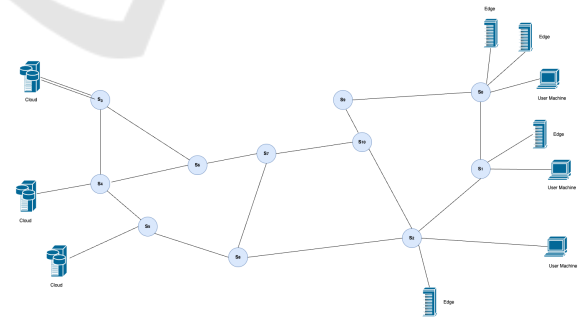


Figure 4: Internet2 Abilene Topology (Beck and Moore, 1998).

The QoS satisfaction of QoS-sensitive applications was considered to evaluate how well the suggested QoS Aware Task Placement (QTP) Algorithm performed. The RYU controller (RYU-Community, 2024) and the Mininet emulator (Mininet-Project, 2024) were used for the evaluation of the algorithm.

The RYU SDN Controller offers application program interfaces (APIs) for creating new control for data flows, while the Mininet emulator allows to simulate a network of virtual switches, hosts, controllers, and links. A directed graph $G = (V, E)$, where V is the set of nodes or switches and E is the set of links, is typically used to depict the underlying network. Based on the acquired link-state, each link (u, v) in E is expressed with QoS information as $C(u, v)$ including bandwidth, etc.

As seen in Figure 4, a network architecture based on the Internet2 Abilene backbone network topology (Beck and Moore, 1998) was also taken into consideration for the assessment. There are eleven Internet service providers (ISPs) in the customized topology. The bidirectional link between the ISPs is configured as of a 10 Mbps capacity, a random packet loss percentage between 1 and 5, and a random delay between 1 and 5 ms. The scenario consists of three cloud-based infrastructure as a service (IaaS) servers linked to respective ISPs (S3, S4, S5). In this experiment, three different kinds of virtual machines (VMs) are taken into consideration in the cloud. Table 1 displays the configuration details for each type of virtual machine. These virtual machine types vary depending on the virtual CPU, memory, and network bandwidth that are available. For instance, LowConfVM provides 1 virtual CPU, 500 MB memory, limited network bandwidth, and elastic storage, while HighConfVM gives 2 virtual CPUs, 3 GB of memory, and high network bandwidth. Each type of virtual machine (VM) has an hourly fee that is determined by the pricing model provided by the Amazon EC2 cloud (Services, 2024). Additionally, the three distinct users are connected via S0, S1, and S2 ISPs while edge devices are also offered by those three ISPs to assist clients, if needed.

An experiment was carried out with a specific case scenario and a few important QoS parameters related to an application's preferred data traffic were identified to quantify the performance. The objective is to evaluate the created QTP algorithm's cost and QoS performance to that of non-QTP algorithms. The non-QTP algorithm places compute-intensive workloads on the cloud without taking the workload's quality of service (QoS) into account. To compare the performance of the algorithms, crucial QoS performance metrics — cost, throughput, latency, and packet loss are taken into account. The total data transfer rate of the application's accepted flows, or throughput, is the result of adding up all of the accepted flows. The percentage of packets lost while the packets are sent by the accepted flows is measured by packet loss. The amount of time taken for a network communication

by a packet is characterized as latency or network delay.

QoS performances are traced for both QTP and non-QTP methods because of the network flows that are observed between a randomly selected client and server. In this experiment, the eHealth workload between client and server was generated using the Iperf (Hardin et al., 2023) program, and the average performance data of many runs was measured appropriately. Reducing latency and packet loss during network communication is crucial for remote healthcare monitoring applications, since their workloads are highly susceptible to changes in throughput, delay, and packet loss (Ravi et al., 2024). The QTP algorithm allows and reprovisions resources (after a fixed interval) between cloud and edge devices based on the anticipated workload. According to the simulation results, the QTP algorithm can successfully increase throughput while lowering costs, packet loss, and network latency for the eHealth application because it can provide the proper environment to meet all necessary QoS requirements. Alternatively, the Non-QTP method does not take into account the anticipated workload i.e. places the application workload statically without considering continuing changes of the workload. Non-QTP cannot satisfy the QoS characteristics (such as latency and loss) during application execution and cannot offer effective resource provisioning for fluctuating workloads. As QTP chose a superior and desired resource provisioning strategy, it subsequently offered greater throughput, fewer packet loss, and less latency as compared to Non-QTP (see Figure 5).

4.1 Results

The tasks of remote healthcare monitoring application exhibit fluctuating workloads over time, predicted using the developed time-series prediction model. Initially, utilizing the developed provisioning algorithm, all tasks were allocated across Cloud-Edge devices based on their QoS characteristics and Compute Sensitivity. With a defined time interval, the QTP algorithm analyzed the future workload of each task, resulting in the identification of the tasks to be relocated between cloud platform and edge devices.

In the experiment, three different predicted time periods are considered i.e. 3 hours, 6 hours, and 12 hours. An increase in throughput was observed when the QoS of running tasks on the Cloud-Edge was assessed for each interval (see Figure 5). Throughput increased considerably as a result of the QTP algorithm's ability to make efficient reallocation strategies. This was made possible by the increased options

Table 1: VM Configuration.

Different Types of VM in Cloud and Related Costs					
VM Type	VCPU	Memory	Network	Storage	Hourly Cost (in Dollar)
HighConfVM	2	2 GB	High	Elastic	0.0188
MedConfVM	1	1 GB	Moderate	Elastic	0.0116
LowConfVM	1	0.5 GB	Low	Elastic	0.0058

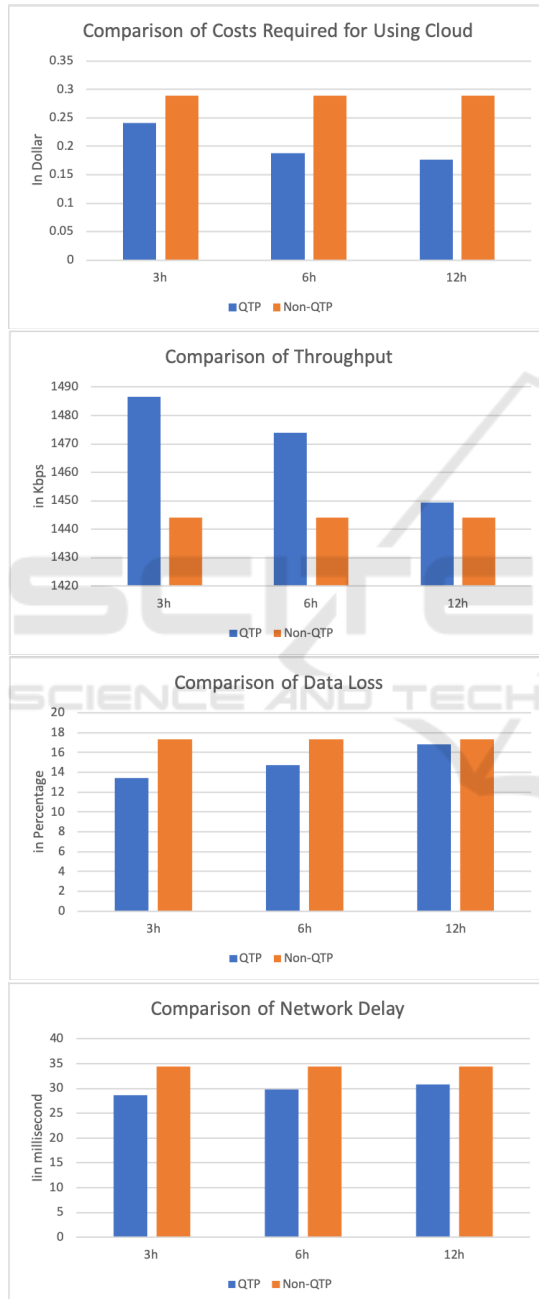


Figure 5: Performance Comparison of Various Algorithms.

for applications to assign tasks to the optimal locations between the cloud and the edge. This advantage is more pronounced when prediction and related rearrangement occur more often, for example, a 3-hours prediction performs better than one made over longer time intervals, i.e. six hours or twelve hours. Again, for each type of prediction interval, latency or network delay, was measured and the results indicate that the more frequently the QTP algorithm validates the relocation plan, the lower the latency (see Figure 5). QTP algorithm also led to a significant decrease in data loss in a similar fashion (see Figure 5). Finally, there are monetary costs associated with executing tasks in cloud environment. With the efficient resource provisioning strategy of QTP algorithm, the duration of tasks execution in cloud is optimized so the cost is also reduced significantly while compared with Non-QTP algorithm (see Figure 5). The experiment results show that the QoS Aware Resource Provisioning in Edge-Cloud can effectively handle the QoS requirements of eHealth application and the benefit is more significant when the tasks are rearranged more frequently based on forecasting model.

5 CONCLUSION

This research addresses the complex challenge of managing Quality of Service (QoS) requirements in Edge-Cloud computing environments, where both Cloud and Edge/Fog computing platforms play crucial roles in data processing tasks. While Edge/Fog computing excels in latency-sensitive applications, scalability remains a concern. The proposed scalable model offers a strategic approach to resource allocation, considering the specific needs of data processing tasks and balancing QoS requirements effectively. By developing an efficient heuristic algorithm and integrating a predictive model for eHealth workload behavior, this research significantly enhances the efficiency and effectiveness of resource management in Edge-Cloud environments. Through simulations, the proposed approach demonstrates superior performance in terms of cost-effectiveness, response time, and resource utilization compared to existing methods. Overall, this research contributes valuable insights into optimizing service delivery and enhancing

user satisfaction in cloud and edge computing ecosystems, paving the way for more robust and scalable applications in the future.

ACKNOWLEDGMENTS

The related work section was paraphrased using ChatGPT (OpenAI, 2024).

FUNDING STATEMENT

This work was funded by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, under the Deanship of Research (Grant-EC213004).

REFERENCES

- Ahmed, S., Irfan, S., Kiran, N., Masood, N., Anjum, N., and Ramzan, N. (2023). Remote health monitoring systems for elderly people: a survey. *Sensors*, 23(16):7095.
- Beck, M. and Moore, T. (1998). The internet2 distributed storage infrastructure project: An architecture for internet content channels. *Computer Networks and ISDN systems*, 30(22-23):2141–2148.
- Chi, H. R., Domingues, M. F., and Radwan, A. (2020). Qos-aware small-cell-overlaid heterogeneous sensor network deployment for ehealth. In *2020 IEEE SENSORS*, pages 1–4. IEEE.
- da Rosa Righi, R., Correa, E., Gomes, M. M., and da Costa, C. A. (2020). Enhancing performance of iot applications with load prediction and cloud elasticity. *Future Generation Computer Systems*, 109:689–701.
- Etemadi, M., Ghobaei-Arani, M., and Shahidinejad, A. (2020). Resource provisioning for iot services in the fog computing environment: An autonomic approach. *Computer Communications*, 161:109–131.
- Hardin, B., Comer, D., and Rastegarnia, A. (2023). On the unreliability of network simulation results from mininet and iperf. *International Journal of Future Computer and Communication*, 12(1).
- Herrera, J. L., Bellavista, P., Foschini, L., Galán-Jiménez, J., Murillo, J. M., and Berrocal, J. (2020). Meeting stringent qos requirements in iiot-based scenarios. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE.
- Hoseiny, F., Azizi, S., Shojafar, M., and Tafazolli, R. (2021). Joint qos-aware and cost-efficient task scheduling for fog-cloud resources in a volunteer computing system. *ACM Transactions on Internet Technology (TOIT)*, 21(4):1–21.
- Hoseinyfarahabady, M. R., Tari, Z., and Zomaya, A. Y. (2019). Disk throughput controller for cloud data-centers. In *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 404–409. IEEE.
- Islam, M. Z., Sagar, A. S., and Kim, H. S. (2024). Enabling pandemic-resilient healthcare: Edge-computing-assisted real-time elderly caring monitoring system. *Applied Sciences*, 14(18):8486.
- Külzer, D. F., Kasparick, M., Palaos, A., Sattiraju, R., Ramos-Cantor, O. D., Wieruch, D., Tchouankem, H., Göttisch, F., Geuer, P., Schwarzmänn, J., et al. (2021). AI4Mobile: Use cases and challenges of AI-based QoS prediction for high-mobility scenarios. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–7. IEEE.
- Louvros, S., Paraskevas, M., and Chrysikos, T. (2023). QoS-aware resource management in 5G and 6G cloud-based architectures with priorities. *Information*, 14(3):175.
- Mininet-Project (2024). Mininet. <https://mininet.org/>.
- Mukhopadhyay, A., Remanidevi Devidas, A., Rangan, V. P., and Ramesh, M. V. (2024). A QoS-aware IoT edge network for mobile telemedicine enabling in-transit monitoring of emergency patients. *Future Internet*, 16(2):52.
- OpenAI (2024). ChatGPT: Conversational AI Model. <https://openai.com/chatgpt>. Accessed: [DATE].
- Peng, D., Sun, L., Zhou, R., and Wang, Y. (2023). Study QoS-aware fog computing for disease diagnosis and prognosis. *Mobile Networks and Applications*, 28(2):452–459.
- Ravi, K. C., Kavitha, G., Prasad, L. H., Srinivasa Rao, N. V., Deivasigamani, S., Ramesh, J. V. N., and Siddiqui, S. T. (2024). Beyond 5g-based smart hospitals: Integrating connectivity and intelligence. *Smart Hospitals: 5G, 6G and Moving Beyond Connectivity*, pages 169–193.
- Rema, V. and Sikdar, K. (2021). Time series modelling and forecasting of patient arrivals at an emergency department of a select hospital. In *Recent trends in signal and image processing: ISSIP 2020*, pages 53–65. Springer.
- RYU-Community (2024). Ryu sdn framework. <https://ryu-sdn.org/>.
- Services, A. W. (2024). Aws pricing calculator for EC2 enhancements. <https://calculator.aws/>.
- Swedish Society for Industrial Organization (2025). Swedish society for industrial organization. <https://ssio.se/>.
- Xu, Y., Li, J., Lu, Z., Wu, J., Hung, P. C., and Alelaiwi, A. (2020). Arvmec: adaptive recommendation of virtual machines for IoT in edge-cloud environment. *Journal of Parallel and Distributed Computing*, 141:23–34.